Specification in PDL with Recursion

Xinxin Liu and *Bingtian Xue*

xinxin,xuebt@ios.ac.cn
Laboratory of Computer Science
Chinese Academy of Sciences
Beijing, China

Motivation

- specification logics
  - Propositional Dynamic Logic (PDL)
    - regular expressions
    - less expressive, but easy to understand
  - modal μ-calculus
    - fixed point
    - more expressive, but hard to understand and analyze: for example, repetition of traces has to be encoded into complex recursive properties

Our Work — Overview

- Extend PDL with simple maximum Fixed Points (disallow alternation) $\longrightarrow$ PDL with recursion (rPDL)
  - Expressive power:
    (regular) PDL $\longrightarrow$ rPDL $\longrightarrow$ μ-calculus
    CTL, CTL* $\longrightarrow$ rPDL with nesting
  - For structured LTS, rPDL (with nesting) is decomposable
  - Satisfiability problem: decidable in EXP-time w.r.t the size of the formula;
    For simple formulas: polynomial in the size of the programs, exponential in the number of the sub-formulas.
  - Solving weak (branching) bisimulation equations of processes

Propositional Dynamic Logic (PDL)

- syntax of PDL
  - propositions or formulas
    1. $\mathrm{tt}, \mathrm{ff} \in \Phi$;
    2. if $\varphi, \psi \in \Phi$ then $\varphi \wedge \psi$, $\varphi \vee \psi \in \Phi$;
    3. if $\varphi \in \Phi$, $\alpha \in \Pi$, then $\langle\alpha\rangle\varphi$, $[\alpha]\varphi \in \Phi$.
  - programs
    1. $Act \subseteq \Pi$;
    2. if $\varphi \in \Phi$ then $?\varphi \in \Pi$;
    3. if α, $\beta \in \Pi$ then $\alpha \cup \beta$,$\alpha; \beta \in \Pi$;
    4. if $\alpha \in \Pi$ then $\alpha^* \in \Pi$.

Enhance PDL $\longrightarrow$ PDL with recursion (rPDL)

- Syntax
  - formula

    4. $X, \overline{X} \in \Phi$ (property identifier)

  - Declaration (Definition)

    $$D \quad ::= \quad \{X_1 = \varphi_1, \ldots, X_m = \varphi_m\}$$

    - No variable is defined more than once in D
    - Well-defined
      D is well defined if $\varphi_1, \ldots, \varphi_m$ are positive formulas w.r.t $X_1, \ldots, X_m$
    - Well-defined declaration has maximum fixed point — exists the weakest property

Enhance PDL $\longrightarrow$ PDL with recursion (rPDL)

- Examples
  - $X = \langle(\tau.?X)^*.a\rangle Y$
    $Y = \langle(\tau.?Y)^*.b\rangle X$
    $\cdot \xrightarrow{\tau} \cdot \cdots \xrightarrow{\tau} \cdot \xrightarrow{a} \cdot \xrightarrow{\tau} \cdots \xrightarrow{\tau} \cdot \xrightarrow{b} \cdot \xrightarrow{\tau} \cdots \xrightarrow{\tau}$
    $X \quad X \quad \cdots \quad X \quad Y \quad \cdots \quad Y \quad X \quad \cdots$
    $\cdot \xrightarrow{a} \cdot \xrightarrow{\tau} \cdots \xrightarrow{\tau} \cdot \xrightarrow{b} \cdot \xrightarrow{\tau} \cdots\cdots\cdots$
    $X \quad Y \quad \cdots \quad Y \quad X \quad \cdots$

Semantics of rPDL

- Semantics of formulas for a given environment ρ
  1. $p \models_\rho \mathrm{tt}$ holds for all $p \in \mathbf{S}$;
  2. $p \models_\rho \mathrm{ff}$ never holds;
  3. $p \models_\rho X$ iff $p \in \rho(X)$;
  4. $p \models_\rho \overline{X}$ iff $p \notin \rho(X)$;
  5. $p \models_\rho \varphi \wedge \psi$ iff $p \models_\rho \varphi$ and $p \models_\rho \psi$;
  6. $p \models_\rho \varphi \vee \psi$ iff $p \models_\rho \varphi$ or $p \models_\rho \psi$;
  7. $p \models_\rho \langle\alpha\rangle\varphi$ iff there exists $q \in \mathbf{S}$ such that $(p, \alpha) \Longrightarrow_\rho q$ and $q \models_\rho \varphi$;
  8. $p \models_\rho [\alpha]\varphi$ iff whenever $(p, \alpha) \Longrightarrow_\rho q$ then $q \models_\rho \varphi$.

Semantics of rPDL

- Semantics of programs for a given environment ρ
  1. $(p, a) \Rightarrow q$ iff $p \xrightarrow{a} q$;
  2. $(p, ?\varphi) \Rightarrow q$ iff $p = q$ and $p \models \varphi$;
  3. $(p, \alpha \cup \beta) \Rightarrow q$ iff $(p, \alpha) \Rightarrow q$ or $(p, \beta) \Rightarrow q$;
  4. $(p, \alpha; \beta) \Rightarrow q$ iff there exists r with $(p, \alpha) \Rightarrow r$ and $(r, \beta) \Rightarrow q$;
  5. $(p, \alpha^*) \Rightarrow q$ iff there exist $n \geq 0, q_0, \ldots, q_n$ such that $(q_i, \alpha) \Rightarrow q_{i+1}$ for $0 \leq i \leq n-1$ and $p = q_0, q_n = q$.

Semantics of rPDL

- Semantics of D for a given environment ρ
  $D = \{X_1 = \varphi_1, \ldots, X_m = \varphi_m\}$ defines an environment for the identifiers:
  $\rho_{\max} = \nu\sigma.\rho\{[\![\varphi_1]\!]\sigma/X_1, \ldots, [\![\varphi_m]\!]\sigma/X_m\}$, where
  $[\![\varphi]\!]\sigma = \{p \mid p \models_\sigma \varphi\}$
  (There exists a unique maximal environment because of the monotonicity.)
- $p \models_D \varphi$ if $p \models_{\rho_{\max}} \varphi$
  $(p, \alpha) \Rightarrow_D q$ if $(p, \alpha) \Rightarrow_{\rho_{\max}} q$

Expressiveness of rPDL

- rPDL $\longrightarrow$ modal μ-calculus
- expressiveness of simple formulas

$*$ CTL,CTL* $\longrightarrow$ rPDL with nesting will be shown elsewhere

Expressiveness of rPDL: rPDL to μ-calculus

- Syntax of μ-calculus (the version that allows simultaneous mutual recursive definitions)

$$
\begin{array}{lcl}
F & ::= & \mathrm{tt} \mid \mathrm{ff} \mid X \mid \overline{X} \mid F \wedge G \mid F \vee G \mid \langle a \rangle F \mid [a]F \\
 & & \mid \text{letmax } D \text{ in } F \mid \text{letmin } D \text{ in } F \\
D & ::= & X_1 = F_1, \ldots, X_n = F_n
\end{array}
$$

Expressiveness of rPDL: rPDL to μ-calculus

- Translation

$$
\begin{array}{lcl}
\mathcal{T}(\varphi) & = & \varphi \quad \text{when } \varphi \text{ is } \mathrm{tt}, \mathrm{ff}, X, \overline{X} \\
\mathcal{T}(\varphi \wedge \psi) & = & \mathcal{T}(\varphi) \wedge \mathcal{T}(\psi) \quad \mathcal{T}(\varphi \vee \psi) \;=\; \mathcal{T}(\varphi) \vee \mathcal{T}(\psi) \\
\mathcal{T}(\langle a \rangle \varphi) & = & \langle a \rangle \mathcal{T}(\varphi) \qquad \mathcal{T}(\langle ?\psi \rangle \varphi) \;=\; \mathcal{T}(\psi) \wedge \mathcal{T}(\varphi) \\
\mathcal{T}(\langle \alpha \cup \beta \rangle \varphi) & = & \mathcal{T}(\langle \alpha \rangle \varphi) \vee \mathcal{T}(\langle \beta \rangle \varphi) \\
\mathcal{T}(\langle \alpha ; \beta \rangle \varphi) & = & \mathcal{T}(\langle \alpha \rangle \langle \beta \rangle \varphi) \\
\mathcal{T}(\langle \alpha^* \rangle \varphi) & = & \text{letmin } Y = \mathcal{T}(\varphi) \vee \mathcal{T}(\langle \alpha \rangle Y) \text{ in } Y \\
\mathcal{T}([a] \varphi) & = & [a] \mathcal{T}(\varphi) \qquad \mathcal{T}([?\psi] \varphi) \;=\; \mathcal{T}(\overline{\psi}) \vee \mathcal{T}(\varphi) \\
\mathcal{T}([\alpha \cup \beta] \varphi) & = & \mathcal{T}([\alpha] \varphi) \wedge \mathcal{T}([\beta] \varphi) \\
\mathcal{T}([\alpha ; \beta] \varphi) & = & \mathcal{T}([\alpha][\beta] \varphi) \\
\mathcal{T}([\alpha^*] \varphi) & = & \text{letmax } Y = \mathcal{T}(\varphi) \wedge \mathcal{T}([\alpha] Y) \text{ in } Y
\end{array}
$$

Expressiveness of rPDL: rPDL to μ-calculus

- **Theorem**:
  $p \models_D \varphi$ if and only if $p \in \mathrm{F}[\![\text{letmax } D^\mu \text{ in } \mathcal{T}(\varphi)]\!]\rho_0$,
  - ρ_0 is an empty environment;
  - $D^\mu = \{X_1 = \mathcal{T}(\varphi_1), \ldots, X_m = \mathcal{T}(\varphi_m)\}$, if $D = \{X_1 = \varphi_1, \ldots, X_m = \varphi_m\}$.

Expressiveness of rPDL: rPDL to μ-calculus

- Examples
  - CTL*: *EGFp*
    μ-calculus: $\nu X.\mu Y.\langle\bullet\rangle((X \wedge p) \vee Y)$
    rPDL: $X = \langle\bullet^*.?p\rangle X$
    $\cdot \longrightarrow \cdots\cdots \longrightarrow \cdot \longrightarrow \cdots\cdots \longrightarrow \cdot \longrightarrow \cdots\cdots\cdots\cdots$
    X $\quad$ p, X $\quad$ p, X

Expressiveness of rPDL: simple formulas

- Simple formula
  - Simple formula: no $\overline{X}$, $[\,] \longrightarrow [a]F$
  - Simple declaration D: whenever $X = \varphi \in D$ then φ is simple.
- Simple formulas can express weak bisimulation ($\approx$) and other bisimulation ($\sim, \approx_b, \ldots$) equivalent classes for a finite process p.

Expressiveness of rPDL: simple formulas

- weak bisimulation ($\approx$) equivalent classes
  - $D = \{X_p = \bigwedge_{p \xrightarrow{a} p'} \langle \tau^*.a.\tau^* \rangle X_{p'}$
    $\wedge \bigwedge_{a \in Act} [a] (\bigvee_{p \overset{\hat{a}}{\Longrightarrow} p'} X_{p'}) \mid p \in \mathbf{S}\}$
  - $p \approx q$ if and only if $q \models_D X_p$

Expressiveness of rPDL: simple formulas

- Translating a positive formula into a simple one

$$
\begin{array}{rcl}
\mathcal{S}(\varphi) & = & \varphi \quad \text{when } \varphi \text{ is } \mathrm{tt}, \mathrm{ff}, X \\
\mathcal{S}(\varphi \wedge \psi) & = & \mathcal{S}(\varphi) \wedge \mathcal{S}(\psi) \\
\mathcal{S}(\varphi \vee \psi) & = & \mathcal{S}(\varphi) \vee \mathcal{S}(\psi) \\
\mathcal{S}(\langle \alpha \rangle \varphi) & = & \langle \alpha \rangle \mathcal{S}(\varphi) \\
\mathcal{S}([a]\varphi) & = & [a]\mathcal{S}(\varphi) \\
\mathcal{S}([?\psi]\varphi) & = & \mathcal{S}(\overline{\psi}) \vee \mathcal{S}(\varphi) \\
\mathcal{S}([\alpha \cup \beta]\varphi) & = & \mathcal{S}([\alpha]\varphi) \wedge \mathcal{S}([\beta]\varphi) \\
\mathcal{S}([\alpha;\beta]\varphi) & = & \mathcal{S}([\alpha][\beta]\varphi) \\
\mathcal{S}([\alpha^*]\varphi) & = & X_{[\alpha^*]\varphi}
\end{array}
$$

Expressiveness of rPDL: simple formulas

- Translating a positive formula into a simple one
  - if $X = \varphi \in D$ then $X = \mathcal{S}(\varphi) \in D_s$
    if $X_{[\alpha^*]\varphi}$ occurs in a right hand side of a definition in D_s,
    then $X_{[\alpha^*]\varphi} = \mathcal{S}(\varphi) \wedge \mathcal{S}([\alpha]X_{[\alpha^*]\varphi}) \in D_s$
  - **Theorem**:
    $p \models_D \varphi$ if and only if $p \models_{D_\mathcal{S}} \mathcal{S}(\varphi)$
  - Every positive formula has an equivalent simple formula.
    Every well defined declaration has an equivalent simple declaration.

Decision problems

- Satisfiability of positive formulas
  - Translating into simple formulas
  - Decision procedure for satisfiability of simple formulas: consistency set
- Satisfiability of rPDL formulas

Decision problems: decision procedure for satisfiability of simple formulas

- saturated set Γ (set of formulas)
  1. whenever $\varphi \wedge \psi \in \Gamma$ then $\varphi \in \Gamma$ and $\psi \in \Gamma$;
  2. whenever $\varphi \vee \psi \in \Gamma$ then $\varphi \in \Gamma$ or $\psi \in \Gamma$;
  3. whenever $X \in \Gamma$ and $X = \varphi \in D$ then $\varphi \in \Gamma$.
- consistency set $\mathcal{C}$ (set of formula sets): $\mathcal{C} \in 2^{\Phi}, \Gamma \in \mathcal{C}$
  1. Γ is saturated;
  2. whenever $\varphi \in \Gamma$ then $\Gamma \models_{\rho^{\mathcal{C}}} \varphi$

     LTS: $\langle \mathcal{C}, Act, \{\xrightarrow{a} \mid a \in Act\} \rangle$

     $\Gamma \xrightarrow{a} \Gamma'$ if: whenever $[a]\psi \in \Gamma$ then $\psi \in \Gamma'$

     $\rho^{\mathcal{C}}(X) = \{\Gamma \in \mathcal{C} \mid X \in \Gamma\}$

Decision problems: decision procedure for satisfiability of simple formulas

- **Theorem**: Let φ be a simple rPDL formula, D be a simple declaration. Then the following two conditions are equivalent:
  1. there exists a consistency set $\mathcal{C}$ and some $\Gamma \in \mathcal{C}$ such that $\varphi \in \Gamma$;
  2. there exists an LTS $\langle \mathbf{S}, Act, \{\xrightarrow{a} \mid a \in Act\}\rangle$ such that $p \models_D \varphi$ for some $p \in \mathbf{S}$.

Decision problems: decision procedure for satisfiability of simple formulas

- sub-formula of φ, *D*
  - sub(φ)

$$\begin{array}{lcll} \text{sub}(\varphi) & = & \{\varphi\} & \text{where } \varphi = \text{tt}, \text{ff}, X, \overline{X} \\ \text{sub}(\varphi \wedge \psi) & = & \{\psi \wedge \psi\} \cup \text{sub}(\varphi) \cup \text{sub}(\psi) & \\ \text{sub}(\varphi \vee \psi) & = & \{\varphi \vee \psi\} \cup \text{sub}(\varphi) \cup \text{sub}(\psi) & \\ \text{sub}(\langle\alpha\rangle\varphi) & = & \{\langle\alpha\rangle\varphi\} \cup \text{sub}(\varphi) & \\ \text{sub}([a]\varphi) & = & \{[a]\varphi\} \cup \text{sub}(\varphi) & \end{array}$$

  - $\text{sub}(D) = \bigcup\{\text{sub}(\varphi) \mid X = \varphi \in D\}$
- sub(φ) has nothing to do with the size of programs in φ
- the cardinality of sub(φ) is much smaller than that of FL(φ) (the usual Fischer-Ladner Closure of φ)

Decision problems: decision procedure for satisfiability of simple formulas

- **Algorithm**: For a given simple formula φ with a simple declaration D, start from $\mathcal{C} = \{\Gamma \subseteq \mathrm{sub}(\varphi) \cup \mathrm{sub}(D)\}$ and LTS $\langle C, Act, \xrightarrow{a} \rangle$. Do the following steps.
  1. For each $\Gamma \in \mathcal{C}$, check whether is saturated, all of which can be checked locally. If not, delete Γ from $\mathcal{C}$.
  2. Repeat the following until $\mathcal{C}$ does not decrease: If there exists $\Gamma \in \mathcal{C}$, $\exists \psi \in \Gamma$ such that $\Gamma \models_{\rho^{\mathcal{C}}} \psi$ does not hold, delete Γ from $\mathcal{C}$.
- the worst time complexity: exponential in the size of the formulas, but polynomial in the size of the programs

Decision problems: an application of rPDL

- Example: solving the process equation of weak bisimulation equivalence $C(x) \approx p$
  - $C(x) \approx p$ if and only if $C(x) \models_D \varphi_p$
    (by results of simple formulas and simple declaration)
  - $C(x) \models_D \varphi_p$ if and only if $x \models_{D^d} \mathcal{W}(C, \varphi_p)$
    (by results of decomposition property [X. Liu and B.Xue, Decomposition of PDL and its extension])
  - $C(x) \approx p$ if and only if $\mathcal{W}(C, \varphi_p)$ is satisfiable
    (using the decision procedure to decide this)

Decision problems: deciding satisfiability of rPDL formulas

- Similar as the usual decision procedure for PDL
- The worst case time complexity: exponential in the size of the formulas and the programs

Conclusion and future work

- Conclusion
  - rPDL strikes a good balance between expressiveness and ease of analysis;
  - rPDL has a simple decision procedure for simple formulas, which is quite expressive.
- Future work
  - Model checking of rPDL
  - Better decision procedure for satisfiability of rPDL formulas
  - Tools for deciding satisfiability, and moreover equation solver (EQ)

- Thanks.